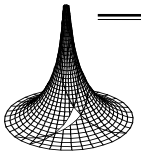

6/2 (2008), 325–343
tmcs@math.klte.hu
http://tmcs.math.klte.hu

Teaching
Mathematics and
Computer Science

Proof step analysis for proof tutoring – a learning approach to granularity

Marvin Schiller, Dominik Dietrich and Christoph Benzmüller

Abstract. We present a proof step diagnosis module based on the mathematical assistant system ΩMEGA. The task of this module is to evaluate proof steps as typically uttered by students in tutoring sessions on mathematical proofs. In particular, we categorise the step size of proof steps performed by the student, in order to recognise if they are appropriate with respect to the student model. We propose an approach which builds on reconstructions of the proof in question via automated proof search using a cognitively motivated proof calculus. Our approach employs learning techniques and incorporates a student model, and our diagnosis module can be adjusted to different domains and users. We present a first evaluation based on empirical data.

Key words and phrases: proof tutoring, automated reasoning, machine learning.

ZDM Subject Classification: U55, E35, E55.

1. Introduction: mathematical assistant system support for teaching proofs

The Dialog project [7] studies natural language-based tutorial dialogue on proofs. Within a tutorial dialogue, the student is given a proof exercise to be solved interactively with the dialogue system. The system provides feedback

This work has been funded by the DFG Collaborative Research Center on Resource-Adaptive Cognitive Processes, SFB 378 (`http://www.coli.uni-saarland.de/projects/sfb378/`), and was supported by a grant from Studienstiftung des Deutschen Volkes e.V.

to the student's solution attempts and aids him in finding a solution, with the overall goal to convey specific concepts and techniques of a given mathematical domain. Due to the flexible and unpredictable nature of tutorial dialogue it is necessary to dynamically process and analyse the informal input to the system, including linguistic analysis of the informal input, evaluation of utterances in terms of soundness, granularity and relevance, and ambiguity resolution at all levels of processing. For the domain reasoning, the DIALOG project employs the mathematical assistant system ΩMEGA [27]. It allows the system to reconstruct students' proof steps [14]. These reconstructions serve as the basis for further analysis, in particular, whether a given step presented by the student is correct, but also whether it is of appropriate step size (i.e. "granularity"), and whether it is relevant.

Overall, a number of different tutorial systems for teaching mathematics have emerged. For propositional and first-order logic there are the CMU proof tutor [26], ProofEasy [11], the HTML-based editor alfie [30], Proofweb [17], Jape [29] and WinKE [12]. For higher-order logic there is ETPS [2]. These systems focus on pure logic and support proof construction using, for example, Fitch-style diagrams or trees. To verify a proof step no search is required. Systems for teaching mathematics at a more abstract level are the EPGY Theorem Proving Environment [28], [20], the Geometry Tutors [18] and Tutch [1]. Those systems allow the user to perform abstract steps and use proof search in a machine-oriented calculus such as resolution to verify them. Huang [16] argues in favor of the *assertion level* as a suitable abstraction layer to represent human proofs. Assertion level proofs justify proof steps by the application of axioms, definitions, or theorems, or on the proof level, such as "by analogy". The notion of assertion level proof competes with other human-oriented calculi, such as the natural deduction calculus [15] and its more refined variations (e.g. PSYCOP [24]). A recent investigation [25] into the correspondence between human proofs and their counterparts in natural deduction points out a mismatch with respect to their granularity. Similar observations have been reported for EPGY in [20], where the resolution- and paramodulation-based theorem prover *Otter* was used to reconstruct human proofs. Limiting the use of *Otter* to a fixed time interval – in order not to allow unreasonably large chains of thinking – turned out to be uninformative of whether a given human proof step was indeed perceived as too complex or not.

In this paper, we argue for an approach that employs assertion level proof search to reconstruct human-made proof steps in the system. Proofs at the assertion level enable the dialogue system to suitably analyse the granularity of human

proof steps, which in turn provides useful information for determining an appropriate reaction of the dialogue system to the student. The structure of this paper is as follows: In Section 2 we present an empirical study which illustrates the role of granularity in tutorial dialogues about proofs. In Section 3 we briefly present the mathematical assistant system ΩMEGA, which is the basis for analysing proof steps in the DIALOG project. In Section 4 we identify criteria that are relevant for determining different levels of granularity. In Sections 5 and 6 we present our granularity analysis module and some first results obtained with a corpus of tutorial dialogues.

2. Evidence from an experiment corpus

Research in the DIALOG project is guided by empirical studies [7], [8], which include two studies in the Wizard-of-Oz paradigm, where human experts (with the help of a special computer interface [9]) simulated the behaviour of a tutoring system for mathematical proofs. These studies highlight the requirements for the modules of the system under development, including the analysis tasks that have to be mastered by the domain reasoner. While the first series of experiments led to the identification of the different domain reasoning tasks, the second series of experiments required the tutors to annotate all domain contributions from the students with judgements concerning correctness (i.e., *correct*, *partially correct*, or *incorrect*), granularity (i.e., *too detailed*, *appropriate*, or *too coarse-grained*) and relevance (i.e., *relevant*, *limited relevance* or *irrelevant*). Both experiments were conducted with students from Saarland University and four experts with teaching experience as the wizards[1]. The exercises were taken from the domains of naive set theory (first experiment series) and binary relations (second experiment series). The second series of experiments involved 37 students, who spent approximately two hours each during an experiment session (including an introduction phase, interaction with the Wizard-of-Oz system and questionnaires).

2.1. Correctness

Most importantly, proof step analysis includes the task of checking whether a proof step is logically correct. An example of such a situation is given in Figure 1.

[1] The experts consisted of the lecturer of a course *Foundations of Mathematics*, a maths teacher, and two maths graduates with teaching experience.

`student_1:`	$(x, y) \in (R \circ S)^{-1}$
`tutor:`	`Now try to draw inferences from that!`
	correct | appropriate | relevant
`student_2:`	`hence` $(y, x) \in (S \circ R)$
`tutor:`	`This step is not correct!`
	incorrect | - | -

Figure 1. Dialogue fragment exhibiting an incorrect step

Correctness – in contrast to granularity and relevance – is relatively simple to verify. EPGY [20], for example, employs proof search in *Otter*. Our solution to checking proof steps employs assertion level proof search and will be presented in Section 3.

2.2. Granularity

Tutors in the Wizard-of-Oz studies were observed to reject proof steps for other reasons than correctness. An example is the dialogue fragment displayed in Figure 2, where the student's task is to show that in the domain of binary relations (where $\circ$ denotes relation composition, and $^{-1}$ denotes inversion), the equality $(R \circ S)^{-1} = S^{-1} \circ R^{-1}$ holds.

`student_1:`	$(x, y) \in (R \circ S)^{-1}$
`tutor:`	`Now try to draw inferences from that!`
	correct | appropriate | relevant
`student_2:`	$(x, y) \in S^{-1} \circ R^{-1}$
`tutor:`	`One cannot directly deduce that.`
	`You need some intermediate steps!`
	correct | too coarse-grained | relevant

Figure 2. Dialogue fragment exhibiting inappropriate step size

The student (tacitly) tries to show that $(R \circ S)^{-1} \subseteq S^{-1} \circ R^{-1}$ by assuming that $(x, y) \in (R \circ S)^{-1}$ (marked as `student_1` in Figure 2). However, the tutor notices that the statement `student_2` requires further elaboration. He explicitly asks the student to subdivide this step into intermediate steps (and indeed, this step is not completely obvious, since it is *not* the case that $(R \circ S)^{-1} = R^{-1} \circ S^{-1} = S^{-1} \circ R^{-1}$, which relies on the misconception that $\circ$ is commutative, which it is

not). However, if we restricted proof step analysis to correctness checking, we would fail to detect any difference between this student step and other more trivial steps.

The developers of the EPGY theorem proving environment [20] encountered similar problems when they used the automated theorem prover *Otter* to check conjectured proof steps from the user. The use of *Otter* was restricted to five seconds, in order not to allow too large "leaps of logic". Still, this allowed *Otter* to sometimes accept seemingly large steps, whereas seemingly easy steps were sometimes not validated. This shows that counting the seconds of using *Otter* is not a suitable measure for granularity.

We develop a more refined measure for granularity in Section 4.

2.3. Relevance

Proof steps which did not advance the proof state with respect to the proof goal were often identified as "irrelevant" by the tutors, for example the step displayed in Figure 3.

`student_2:` $(a, b) \in S^{-1} \Leftrightarrow (b, a) \in S$
`tutor:` `This step is not relevant`

correct	appropriate	irrelevant

Figure 3. Dialogue step lacking relevance (for proof problem: $(R \circ S)^{-1} = S^{-1} \circ R^{-1}$)

Relevance, like granularity, is a challenging topic for the dialogue-based teaching of proofs. In the remainder of this paper we focus on the problem of identifying appropriate levels of granularity of human-made proof steps. We address the problem by using proof constructions at the assertion level, as supported in the ΩMEGA system.

3. The domain reasoner: ΩMEGA

The DIALOG project employs the mathematical assistant system ΩMEGA [27] (i) to represent the mathematical theory in which the proof exercise is carried out, that is, definitions, axioms, and theorems of a certain domain (ii) to represent the ongoing proof attempts of the student using ΩMEGA's proof data structure [4], and (iii) to dynamically reconstruct intermediate steps necessary to verify each

step entered by the student (see [14]). This allows us to support tutoring in the spirit of *cognitive constructivism* [19], such that for a given proof problem, a large variety of solutions can be reconstructed and analysed. In particular, given a proof problem including the to-be-proven statement and the required definitions and facts, any valid deduction up to a predefined number of assertion level steps can be reconstructed (cf. [14]). These reconstructed proofs serve as the basis for the further analysis of the students' proof steps with respect to correctness, granularity and relevance.

Different from other approaches to automated theorem proving, ΩMEGA uses an *assertion application* mechanism [13], which is based upon Serge Autexier's CoRe calculus [3], as its logical kernel. The notion of assertion level proofs is due to Huang [16], and characterises a proof representation where all inference steps are justified by a mathematical fact from the knowledge base, such as definitions, theorems and lemmata. Whereas originally, the assertion level was only the target language for the presentation of machine-generated proofs (e.g. in a natural deduction calculus), ΩMEGA now directly constructs proofs at the assertion level.

CoRe and our assertion level inference mechanism are (higher-order) variants of the deep inference approach[2], that is, they support deductions deeply inside a given formula without requiring preceding structural decompositions as needed in natural deduction (or sequent calculus). As a result, we obtain proofs where each inference step is justified by a mathematical fact, such as a definition, a theorem or a lemma. To illustrate the difference between a typical proof step from a textbook and its formal counterpart in natural deduction consider the following example:

Given the definition of subset

$$\forall U, V \,.\, U \subset V \Leftrightarrow \forall x.x \in U \Rightarrow x \in V$$

an assertion step consists of deriving $a_1 \in V_1$ from $U_1 \subset V_1$ and $a_1 \in U_1$. The corresponding natural deduction proof is shown below:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\forall U, V.\, U \subset V \Leftrightarrow \forall x.x \in U \Rightarrow x \in V}{\forall V.\, U_1 \subset V \Leftrightarrow \forall x.x \in U_1 \Rightarrow x \in V}\,\forall_E}{U_1 \subset V_1 \Leftrightarrow \forall x.x \in U_1 \Rightarrow x \in V_1}\,\forall_E}{U_1 \subset V_1 \Rightarrow \forall x.x \in U_1 \Rightarrow x \in V_1}\,\Leftrightarrow_E \qquad U_1 \subset V_1}{\forall x.x \in U_1 \Rightarrow x \in V_1}\,\Rightarrow_E}{a_1 \in U_1 \Rightarrow a_1 \in V_1}\,\forall_E \qquad a_1 \in U_1}{a_1 \in V_1}\,\Rightarrow_E$$

[2] `http://alessio.guglielmi.name/res/cos/index.html`

Even though natural deduction proofs are far more readable than proofs in machine-oriented formalisms such as resolution, we see that they are at a much lower level than proofs typically found in mathematical textbooks. In the example above, a single assertion step corresponds to six steps in the natural deduction calculus. This is mainly because each natural deduction rule stands for a simple manipulation of the logical structure of a formula. Assertion level inference rules in ΩMEGA are automatically generated from the axioms of the problem statement (cf. [5]).

The reconstruction of a student proof step in ΩMEGA is achieved by using a *depth-limited breadth-first search* (with pruning of superfluous branches). For a given proof state and one utterance, all possible successor states up to a specified depth limit are constructed. From these, those successor states that match the given utterance with respect to some filter function (analysing whether a successor state is a possible reading of the student proof step) are selected. An utterance that leads to at least one such successor state is reported by the module to be correct, otherwise it is reported to be incorrect. It is possible that a proof step is wrongly rejected because of a too restrictive depth limit. However, a first case study shows that even with a depth limit of four assertion level steps, the vast majority of steps (95.6%) taken from a sample of proofs obtained in the second Wizard-of-Oz experiment can be correctly identified as correct or incorrect (cf. [6]).

4. Granularity criteria

In order to develop an algorithm that judges the granularity of individual (human-made) proof steps, we have started with the simple approach of reconstructing these proof steps in a suitable calculus (which generally resulted in several calculus level proof steps corresponding to one single utterance), and identifying the step size of a given proof step with the number of calculus level proof steps that correspond to it. A case study (reported in [25]) was undertaken with both Gentzen's natural deduction calculus [15] and the psychologically motivated PSYCOP calculus [24] as the base calculus for the proof reconstruction. However, the study provided evidence that counting calculus level steps in neither of the two calculi provided a sufficient means for characterising granularity, and that more sophisticated criteria for measuring granularity are required. In particular, the approach did not account for all the different granularity-related phenomena

we could observe in the corpora obtained in the Wizard-of-Oz experiments. We list some of them below.

Merging different concepts

The combination of several applications of the same definition or theorem (relating to the same concept) into one proof step was observed frequently, consider the example in Figure 4.

`student_4:`	$\exists$ `z , such that` $(b,z) \in R$ `and` $(z,a) \in S$	
`tutor:`	`Right.`	correct | appropriate | relevant
`student_5:`	`Then` $(z,b) \in R^{-1}$ `and` $(a,z) \in S^{-1}$	
`tutor:`	`Correct.`	correct | appropriate | relevant

Figure 4. Dialogue fragment illustrating two applications of the concept of relation inverse

Here, the fact that $(x,y) \in R$ iff $(y,x) \in R^{-1}$ for any x,y is applied twice. This was never subject to criticism by the tutors. In fact, from a cognitive viewpoint, applying the same mathematical fact several times requires retrieving the relevant concept in memory only once, where it is readily available for subsequent applications. The same is not true for using several different concepts in one step, which was sometimes subject of criticism from the tutors. Therefore, we consider the number of different concepts required to justify a given proof step as one criterion for its granularity (rather than the mere number of calculus level steps).

Note that this is easily possible in our approach, as we reconstruct and maintain a student's proof attempt at the assertion level. Here, a single deduction step corresponds to a concept application. Consequently the information about what concepts are involved is directly available. This this not the case in natural deduction or resolution, where this information is generally more difficult to obtain.

Verbal explanation

Whether students explicitly referred to the concepts that they used in their proofs was also a criterion for the tutors in the experiment. Consider the dialogue fragment in Figure 5, where the tutor considers the step leading to utterance `student_9` as too coarse-grained *unless* the student provides further verbal evidence that he can justify this step. Therefore, when judging about granularity, it

is of interest to consider how many (and probably which) concepts were applied in a student's step without mentioning them verbally.

student_8:	However, this means: $(z, y) \in R^{-1}$ and $(x, z) \in S^{-1}$	
tutor:	Now it is correct.	correct | appropriate | relevant
student_9:	Therefore it follows: $(x, y) \in S^{-1} \circ R^{-1}$, what was to be shown.	
tutor:	Correct. Please give a (simple) justification for this last step of the proof.	
	correct | too coarse-grained | relevant	
student_10:	This follows immediately from the definition of the relation product.	
tutor:	Right. With this, you have solved the exercise.	
	correct | appropriate | relevant	

Figure 5. Dialogue fragment involving verbal explanation

To detect the concepts mentioned by the student we currently employ key words extracted from the student's utterance. However, we plan to integrate a more sophisticated analysis in the near future, based on linguistic investigations within the scope of the DIALOG project.

Introducing hypotheses

In the experiment corpus, steps that introduced new hypotheses (for example, when an implication was shown by assuming the premise, in order to derive the implication's conclusion) generally stood on their own, and were not combined into much larger steps. Indeed, introducing a new hypothesis to the proof can be a crucial step towards the solution. Thus, steps that introduce new assumptions have a special status, they need to be spelled out explicitly.

Introducing subgoals

The corpus also showed that steps which split the current goal into several (independent) subgoals have a special status, and should hence be taken into account for the granularity analysis. For example, splitting the proof of showing set equality between two relations into two directions $\subset$ and $\supset$ is an important step. We also found that students implicitly introduced a subgoal by only stating a hypothesis, as shown below:

student_1:	let $(a, b) \in (R \circ S)^{-1}$	
tutor:	Right.	correct | appropriate | relevant

Here, the proof task is to show that $(R \circ S)^{-1} = S^{-1} \circ R^{-1}$, therefore we may suppose that the student intends to show that $(a, b) \in S^{-1} \circ R^{-1}$ according to the extensionality principle.

Learning progress & student modelling

The Wizard-of-Oz experiments made use of a series of exercises which became gradually more advanced, where previously proven statements could be used in subsequent proofs. The tutors encouraged the students to use these previously proven statements as lemmata, such that they had the same status as those mathematical facts (e.g. definitions, properties) they were initially provided with as an introduction into the mathematical domain. We model this in our dialogue system by making previously applied sequences of proof steps, and of course, the statement of the finished proof, a part of the mathematical theory during the tutorial dialogue, such that each one subsequently becomes an atomic inference rule at the assertion level. Furthermore, we use a student model to keep track of those concepts a student has previously mastered (these facts are recorded during the dialogue) and of those he possibly does not know or has not applied before. This way, a sequence of novel steps can be given a different status with respect to granularity than one that includes only well-known steps. In the following, we consider only the sheer number of concepts that are supposed to be known and supposed to be unknown, respectively, to be relevant for judging granularity.

We do not claim that the above criteria are exhaustive, since they are based on a particular series of experiments in one mathematical domain only. Furthermore, the question remains what weight to give to each of these criteria (for example, does the verbal explanation counterbalance a high number of facts combined into one step?). There is also the question of how to employ the observed criteria for the actual generation of useful feedback to the user, which requires didactic considerations beyond the scope of this paper.

5. Judging granularity

The result of granularity analysis for an uttered proof step is a granularity judgement, which can take one out of three possible values: *appropriate*, *too detailed*, and *too coarse-grained*. This section illustrates how information from the proof reconstructions with respect to the criteria discussed above is used to categorise the to-be-analysed proof steps. Consider again the example proof step

`student_2` presented in Figure 2. ΩMEGA reconstructs the following derivation for the student input, shown in sequent notation (in this example, we assume that the student model considers all concepts as not yet mastered by the student).

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma, (x,y) \in S^{-1} \circ R^{-1} \vdash \Delta}{\Gamma, (z,y) \in R^{-1} \wedge (x,z) \in S^{-1} \vdash \Delta}\ \text{Def}\circ}{\Gamma, (z,y) \in R^{-1} \wedge (z,x) \in S \vdash \Delta}\ \text{Def}^{-1}}{\Gamma, (y,z) \in R \wedge (z,x) \in S \vdash \Delta}\ \text{Def}^{-1}}{\Gamma, (y,x) \in R \circ S \vdash \Delta}\ \text{Def}\circ}{\Gamma, (x,y) \in (R \circ S)^{-1} \vdash \Delta}\ \text{Def}^{-1}$$

From this reconstruction, the following information can be extracted.

Total number of steps: This is simply the number of assertion level inference steps, which yields a value of "5" in our case.

Number of different concepts: Since only two distinct assertions were used (the definitions of the inverse relation $^{-1}$ and relation composition $\circ$), this yields a value of "2" in our case.

Number of previously used concepts: For each employed concept (here, the definitions of $\circ$ and $^{-1}$), we look up in the student model if they are already known to the student. If we assume for our example that this is not the case, we obtain a value of "0".

Number of not previously used concepts: This is simply the difference of the total number of concepts and the number of previously used concepts, in the case of our example this will consequently indicate "2" new concepts.

Verbal explanation: This is extracted from the utterance via natural language analysis. Since in the example, there is no accompanying explanation, we report that "2" concepts rest unexplained.

Introduced hypotheses: None of the above steps introduces a new hypothesis, therefore the result is "0".

Number of introduced subgoals: In the absence of newly introduced subgoals, this also yields a "0".

As a result, we obtain a *granularity observation* tuple, where each entry represents one of the evaluated granularity criteria for the given step: $(5, 2, 0, 2, 0, 0)$.

A simple model, which we have implemented in our approach, is to formulate the correspondence between such evaluation results and the final judgement as

simple *if-then* rules, which provide a mapping between the values of the automatically determined granularity criteria for a proof step and the corresponding granularity labels *appropriate*, *too coarse-grained* and *too detailed*[3]. To formulate these rules, we employ the LISA rule environment (see [31]), which provides the infrastructure for building expert systems. This allows us to formulate rules such as:

IF `number-of-different-concepts` $> 1 \wedge$ `number-of-unexplained-concepts` > 1
THEN `result=too coarse-grained`

This simple rule expresses that whenever we have a proof step that requires two or more different assertions, and two or more of the required assertions are not explained verbally, we classify the proof step as too coarse-grained. However, care has to be taken in those cases where for a given proof step, more than one rule is applicable with conflicting results. For the purpose of conflict resolution, rules can be given different weights in order to decide which rule in the conflict set is given priority. In the following, we consider the learning of decision rules from empirical data.

6. A machine learning approach to granularity

Whereas the proof step evaluation with respect to granularity criteria as described above can easily be computed given the proof reconstructions in ΩMEGA and a description of the verbal input, it is not *a priori* clear how to turn this information into an appropriate final judgement whether the proof step in question is of appropriate step size in the given context or not. The answer to that question may in particular depend on the preferences of a particular human tutor and the particular mathematical domain under consideration. Therefore, we turn the problem into a machine learning problem in which individual preferences and domain dependency in the granularity judgements can be learned. Training instances are pairs of the granularity observation tuples as described in Section 5 together with a corresponding class label in the form of a granularity judgement by the human tutor (one of *appropriate*, *too coarse-grained* and *too detailed*).

Currently, we use the C5.0 data mining tools (see [23] and also [22]) – which support the learning of decision trees and of rule sets – to obtain classifiers for

[3]Ideally, the association between the criteria and the granularity judgements mimics the decisions of the human tutors.

granularity. A learned decision tree can be rewritten straightforwardly to an equivalent rule set, which can be used in the same way as described in the previous section. Nevertheless, rule sets generated by C5.0 can under circumstances be more accurate predictors than decision trees, but these rule sets may not be conflict-free. C5.0 provides confidence values for the learned rules to aid conflict resolution.

Thus, we employ two modules for granularity analysis; one serves to obtain training instances, from which the associations between granularity criteria and granularity judgements can be learned. Using this, a judgement component can then automatically perform granularity judgements. This architecture allows to adapt to the way an individual human tutor makes granularity judgements (and thus to gauge mathematical practice without asking explicit questions), but at the price of requiring previous training of the granularity analysis. Training can be performed either with the help of an annotated corpus of proofs (i.e., where each proof step already carries a granularity label), or in an interactive session with the human expert.

6.1. Setup

An overview on the setup for the training module for learning from an annotated corpus is given in Figure 6.

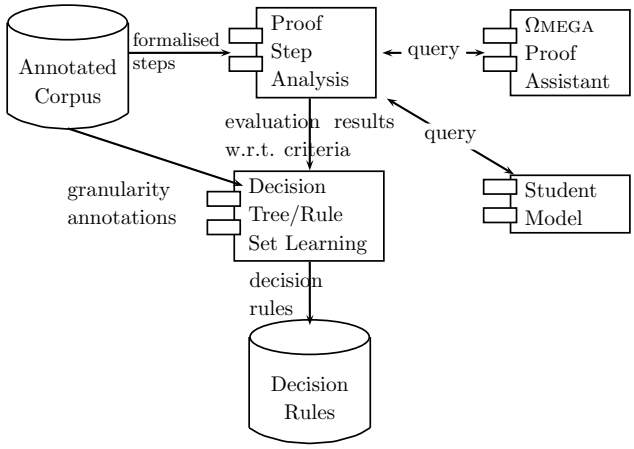

Figure 6. Overview over the architecture for learning granularity judgements from an annotated corpus

Each proof step utterance in the corpus is formally represented in the logical representation language of our system, sent to the analysis module and handed over to the ΩMEGA system for verification. If successful, this yields an assertion level proof, which can be analysed with respect to the granularity criteria with the help of the student model (and possibly also keyword-based verbal content of the proof step under consideration stored in the corpus). The training instances obtained this way are labelled with the corresponding granularity judgements stored in the corpus and handed over to the learning algorithm, which produces a classifier, i.e., a set of decision rules or a decision tree. In the interactive mode, the training module computes the "corpus" on the fly. It steps through a given proof (which can be automatically generated from a problem statement with ΩMEGA, or by hand) at a variable (but bounded) step size in the number of inference steps at the assertion level. That is, for a given proof state, and a random (but bounded) number n, the module proceeds n assertion level inference applications further with the proof and prints the associated formula to the expert who has to provide a granularity judgement (note that the $n-1$ intermediate steps are skipped in the presentation).

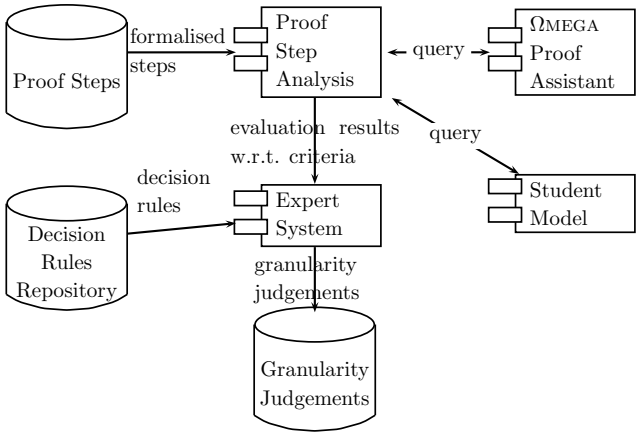

Figure 7. Overview over the architecture for judging granularity

The judgement module is displayed in Figure 7. As part of the proof step analysis, each attempted proof step is handed over to ΩMEGA. In case it can be verified, the resulting assertion level proof is analysed in the light of the student model and a description of the verbal input, yielding a granularity observation tuple with respect to our granularity criteria. We finally use the set of decision rules

previously learned via machine learning to produce the corresponding granularity judgement.

6.2. Evaluation

We have performed an evaluation on a subset of the dialogues from the corpus of the experiments reported in [8]. This subset of dialogues includes only dialogues which contain at least one correct proof step labelled as *too coarse-grained* or *too detailed* by the tutors. This excludes incorrect or partially correct steps, because – unlike human tutors – the system is not able to guess the student's intentions in such a case. The sub-corpus of proof steps we obtain this way includes 47 steps, of which 11 are *too coarse-grained* and one is *too detailed*, the rest is of *appropriate* granularity. Using 10-fold cross validation[4], we achieve a mean classification error of 13% and an inter-rater reliability coefficient κ=0.65 with C5.0 decision tree learning. This is considerably better than naively assigning all steps to be *appropriate*, knowing that the majority is *appropriate*, which would still result in a classification error of 27.5%. Nonetheless, using the classifier SMO [21], which implements a support vector machine, we obtain a mean classification error of $6,4\%$ and κ=0.84.

Even though at first glance, the support vector machine approach classifies better in our example, the decision tree learning approach (as well as learning rule sets) is more informative with respect to the question which of the criteria mentioned above would be most useful for characterising the behaviour of the tutors in the experiments. During the decision tree learning, two (extremely simplistic) decision trees (producing approximately the same error rate) emerged, which we express as rule sets for brevity:

Rule set 1: `IF number-of-not-previously-used-concepts > 1`
`THEN result= too coarse-grained ELSE result= appropriate`

Rule set 2: `IF total-number-of-steps > 3`
`THEN result= too coarse-grained ELSE result= appropriate`

The simplicity of these one-rule rule sets owes to the fact that our sample was rather small. It shows that for the given examples, the role of verbal explanation is negligible (otherwise this criterion would appear in the rules). Also,

[4] We use cross validation since 47 instances are a very small sample indeed. We concentrated on this small sample because collection and processing of empirical data such as in our Wizard-of-Oz experiments is in itself already a very work-intensive process.

knowing the number of different concepts that were employed at once is inferior to simply knowing the total number of inference steps, which provides the more relevant criterion according to the learned rules. Nevertheless, knowing how many of the employed concepts are not familiar to the student according to the student model provides a valid means to distinguish between appropriate and inappropriate proof steps. Note that these observations only apply to the particular experiment sessions reported in [8], which include judgements by different tutors. However, by the virtue of being a learning approach, our granularity analysis can adapt to other domains, teachers, etc.

7. Discussion and conclusion

We have presented an approach to automating granularity judgements for human proof steps, based on the identification of relevant criteria. The assertion level proofs as produced by ΩMEGA are directly amenable to our granularity analysis with respect to these criteria. Furthermore, we have demonstrated how machine learning techniques can be used to obtain context-adapted granularity judgements. The approach can easily be extended to other criteria than the ones exemplified in Section 4. Our method does not require pre-authored solutions for the proof exercises, but makes use of dynamic proof reconstructions at the assertion level in ΩMEGA. Furthermore, the granularity analysis can be trained interactively by human teachers without requiring them to know the internals of the analysis module or to write any classification rules by hand. As shown by our first evaluation, our method provides a means to evaluate how the granularity judgements of a teacher or a group of teachers depend on different criteria (and also, which criteria are negligible). Finally, we have gained some evidence that the student model and reconstructions at the assertion level are useful ingredients for our granularity analysis.

Future work includes a study which is more focused on the evaluation of the granularity analysis module than the previous experiments, for which the interactive training module was not yet available.

8. Acknowledgement

We would like to thank Bruce McLaren for his input and comments on initial drafts of this paper. We thank an anonymous referee for helpful comments.

References

[1] A. Abel, B.-Y. E. Chang and F. Pfenning, Human-readable machine-verifiable proofs for teaching constructive logic, in: *IJCAR Workshop on Proof Transformations, Proof Presentations and Complexity of Proofs (PTP-01)*, Universita degli Studi di Siena, Italy, 2001, 33–48.

[2] P. B. Andrews, C. E. Brown, F. Pfenning, M. Bishop, S. Issar and H. Xi, ETPS: A system to help students write formal proofs, *Journal of Automated Reasoning* **32.** (2004), 75–92.

[3] S. Autexier, The core calculus, in: *Automated Deduction - CADE-20, 20th Int. Conference on Automated Deduction, Tallinn, Estonia, July 22-27, 2005, Proceedings, volume 3632 of LNCS*, Springer, 2005, 84–98.

[4] S. Autexier, C. Benzmüller, D. Dietrich, A. Meier and C.-P. Wirth, A generic modular data structure for proof attempts alternating on ideas and granularity, in: *Mathematical Knowledge Management, 4th International Conference, MKM 2005, Bremen, Germany, July 15-17, 2005, Revised Selected Papers, volume 3863 of LNCS*, (M. Kohlhase, ed.), Springer, 2006, 126–142.

[5] S. Autexier and D. Dietrich, Synthesizing proof planning methods and omega-ants agents from mathematical knowledge, in: *Mathematical Knowledge Management, 5th International Conference, MKM 2006, Wokingham, UK, August 11-12, 2006, Proceedings, volume 4108 of LNCS*, (J. Borwein and B. Farmer, eds.), Springer, 2006, 94–109.

[6] C. Benzmüller, D. Dietrich, M. Schiller and S. Autexier, Deep inference for automated proof tutoring, in: *KI 2007: Advances in Artificial Intelligence. 30th Annual German Conference on AI, volume 4667 of LNAI*, (J. Hertzberg, M. Beetz and R. Englert, eds.), Springer, 2007, 435–439.

[7] C. Benzmüller, A. Fiedler, M. Gabsdil, H. Horacek, I. Kruijff-Korbayová, M. Pinkal, J.H. Siekmann, D. Tsovaltzi, B. Q. Vo and M. Wolska, Tutorial dialogs on mathematical proofs, in: *Proc. IJCAI Workshop on Knowledge Representation and Automated Reasoning for E-Learning Systems*, Acapulco, Mexico, 2003, 12–22.

[8] C. Benzmüller, H. Horacek, H. Lesourd, I. Kruijff-Korbajová, M. Schiller and M. Wolska, A corpus of tutorial dialogs on theorem proving; the influence of the presentation of the study-material, in: *Proc. Int. Conference on Language Resources and Evaluation (LREC-06)*, ELDA, Genoa, Italy, 2006, 1766–1769.

[9] C. Benzmüller, H. Horacek, H. Lesourd, I. Kruijff-Korbayová, M. Schiller and M. Wolska, DiaWOz-II – a tool for wizard-of-oz experiments in mathematics, in: *KI 2006: Advances in Artificial Intelligence, volume 4314 of LNAI*, (C. Freksa, M. Kohlhase, and K. Schill, eds.), Springer, 2006, 159–173.

[11] R. Burstall, *Teaching people to write proofs*, CafeOBJ Symposium, Numazu, Japan, 1998.

[12] M. D'Agostino and U. Endriss, WinKE: a proof assistant for teaching logic, in: *Proc. of the First Int. Workshop on Labelled Deduction (LD'98)*, (D. Basin and L. Vigano, eds.), Freiburg, Germany, 1998.

[13] D. Dietrich, *The task-layer of the Omega system*, Diploma thesis, FR 6.2 Informatik, Universität des Saarlandes, Saarbrücken, Germany, 2006.

[14] D. Dietrich and M. Buckley, *Verification of proof steps for tutoring mathematical proofs*, in: *Proc. Artificial Intelligence in Education (AIED 2007) volume 158.*, (R. Luckin, K. R. Koedinger, and J. Greer, eds.), IOS Press, Los Angeles, California, 2007, 560–562.

[15] G. Gentzen, Untersuchungen über das logische Schliessen, *Math. Zeitschrift* **39** (1934), 176–210, 405–431.

[16] X. Huang, Reconstructing proofs at the assertion level, in: *Automated Deduction – CADE-12, volume 814 of LNCS*, (A. Bundy, ed.), Springer, 1994, 738–752.

[17] C. Kaliszyk, F. Wiedijk, M. Hendriks and F. van Raamsdonk, Teaching logic using a state-of-the-art proof assistant, in: *Proc. Int. Workshop on Proof Assistants and Types in Education*, (H. Geuvers and P. Courtieu, eds.), Paris, France, 2007, 37–50.

[18] K. Koedinger and J. R. Anderson, Reifying implicit planning in geometry: Guidelines for model-based intelligent tutoring system design, in: *Computers as cognitive tools*, (S. P. Lajoie and S. J. Derry, eds.), Erlbaum, Hillsdale, NJ, 1993, 15–46.

[19] C. H. Liu and R. Matthews, Vygotsky's philosophy: Constructivism and its criticisms examined, *Int. Education Journal* **6**, no. 3 (2005), 386–399.

[20] D. McMath, M. Rozenfeld and R. Sommer, A computer environment for writing ordinary mathematical proofs, in: *LPAR, volume 2250 of LNCS*, (R. Nieuwenhuis and A. Voronkov, eds.), Springer, 2001, 507–516.

[21] J. C. Platt, Fast training of support vector machines using sequential minimal optimization, in: *Advances in Kernel Methods – Support Vector Learning*, (B. Schölkopf, C. Burges and A. Smola, eds.), MIT Press, 1999, 185–208.

[22] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.

[23] RuleQuest Research, Data mining tools see5 and c5.0, 2007, `http://www.rulequest.com/see5-info.html`.

[24] L. J. Rips, *The psychology of proof: deductive reasoning in human thinking*, MIT Press, Cambridge, MA, 1994.

[25] M. Schiller, C. Benzmüller and A. Van de Veire, Judging granularity for automated mathematics teaching, in: *LPAR 2006 Short Papers Proc.*, Phnom Phen, Cambodia, 2006.

[26] W. Sieg and R. Scheines, Computer Environments for Proof Construction, *Interactive Learning environments* **4**, no. 2 (1994), 159–169.

[27] J. H. Siekmann, C. Benzmüller and S. Autexier, Computer supported mathematics with Omega, *Journal of Applied Logic* **4**, no. 4 (2006), 533–559.

[28] R. Sommer and G. Nuckols, A Proof Environment for Teaching Mathematics, *Journal of Automated Reasoning* **32**, no. 3 (2004), 227–258.

[29] B. Sufrin and R. Bornat, *User Interfaces for Generic Proof Assistants Part I: Interpreting Gestures*, in: *Proc. of User Interfaces for Theorem Provers (UITP-06)*, York, U.K., 1996.

[30] B. von Sydow, Alfie, a proof editor for propositional logic, 2000, http://www.cs.chalmers.se/~sydow/alfie/index.html.

[31] D. E. Young, The lisa project, 2006, http://lisa.sourceforge.net/.

MARVIN SCHILLER and DOMINIK DIETRICH
DEPARTMENT OF COMPUTER SCIENCE
SAARLAND UNIVERSITY
66041 SAARBRÜCKEN
GERMANY

E-mail: schiller@ags.uni-sb.de

E-mail: dodi@ags.uni-sb.de

CHRISTOPH BENZMÜLLER
DEPARTMENT OF COMPUTER SCIENCE
SAARLAND UNIVERSITY
66041 SAARBRÜCKEN
GERMANY
AND
COMPUTER LABORATORY
THE UNIVERSITY OF CAMBRIDGE
CAMBRIDGE, CB3 0FD
UK

E-mail: chris@ags.uni-sb.de

(Received September, 2007)